UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/606,641	06/29/2000	Kimberly J. Rush	06576-105006	7703

7590 09/10/2003

King & Spalding
191 Peachtree Street N E 45th Floor
Atlanta, GA 30303

EXAMINER	
CHOUDHURY, AZIZUL Q	

ART UNIT	PAPER NUMBER
2143	

DATE MAILED: 09/10/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/606,641	RUSH ET AL.	
	Examiner	Art Unit	
	Azizul Choudhury	2143	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *08 February 2001* .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-20* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-20* is/are rejected.

7)☒ Claim(s) *6* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____ .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

Office Action

Specification

1. The abstract of the disclosure is objected to because the term "to" is believed to be missing in between the terms "system" and "access" within the first sentence. Appropriate correction is required. See MPEP § 608.01(b).

2. The specification of the disclosure is objected to due to lack of TM notations. The terms "Microsoft Word" and "Lotus Notes" should be listed as "Microsoft Word™" and "Lotus Notes™" respectively. Appropriate corrections are required.

Claim Objections

3. Claim 6 is objected to because of the following informalities:

- The term "of" is believed to be missing in between the terms "method" and "claim."
- A space is believed to be missing on page 11, line 15 between the terms "service" and "manager."

4. Appropriate corrections are required.

Claim Rejections - 35 USC § 102

5. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

6. Claims 1-4 are rejected under 35 U.S.C. 102(b) as being anticipated by Doyle et al (US005838906A), referenced hereafter as Doyle.

7. With regards to claim 1, Doyle teaches: a method for accessing multiple types of electronic content, comprising the steps of:

- Requesting a list of available services for a client program module (Doyle discloses a design that features a "list of applications," (column 15, line 14, Doyle). This list of applications is equivalent to the list of services claimed);
- Creating a list of services available to the client program module (Doyle discloses a design that features a "user-defined list of applications," (column 15, line 14, Doyle). The fact that the list is user-defined shows that the list has to be created);
- Selecting a service from the list (Doyle discloses in his design that an application can be launched from the list described above, (column 15, lines 12-14, Doyle). Applications are interpreted as being equivalent to services. The fact that applications can be launched from the list demonstrates that services can be selected from the list);
- Creating a service from a plurality of program objects that are enumerated in the list (Doyle has disclosed that applications can be launched from the list. The launch of an application is equivalent to the creation of a service (column 15, lines 9-21, Doyle)); and
- Providing the client program access to the service, wherein the service manipulates data within the client program module or the service generates some output data that is accessible within the client program module (Doyle's design features a browser client program which serves as an interface on the local user machine to another application/program that could reside on another machine. The non-client program

can access the applications/programs on the other machines that are listed in the list described above. A program is therefore accessed and used through another program. This enables data to be manipulated on the local machine with other services as claimed, (columns 8-10, Doyle)).

8. With regards to claim 2, Doyle teaches: the step of creating a service that further comprises the steps of:

- Obtaining loader identification data and location data for a program object from a local storage medium (Doyle describes a design that has the means by which to identify the application to be launched from the local user client machine, (column 15, lines 18-21, Doyle). Such means are equivalent to loading id and addresses for the program/application to be used, as claimed); and
- Retrieving program object data from a non-local storage medium corresponding to the loader identification data and location data for the program object (As stated before, the client machine can access remote computers in Doyle's design, hence, non-local storage is available. The objects can be on the remote computers and can be accessed; hence a form of loader id must be available as well in the design).

9. With regards to claim 3, Doyle teaches: that the local storage medium comprises a cache memory device (Cache is a "fast storage buffer in the central processing unit of a computer," (American Heritage College Dictionary, 4th Ed). Doyle's design consists of local machines that have RAM, (column 8, lines 38-42, Doyle). It is well known in the art at the time of the invention that computers have a portion of RAM used for cache purposes as claimed).

10. With regards to claim 4, Doyle teaches: that the loader identification data and location data are passed according to one of an exchange method, a set attribute method, and a get attribute method in combination with a variant parameter, whereby arbitrary data content is accessible to the client program module (Doyle's design discloses that a communication protocol is used to exchange information between the browser client and the application, (column 9, lines 7-10, Doyle). This exchange is equivalent to having the claimed data passed. The communication protocol disclosed in Doyle's design is called XEvent which has attributes to set and get as claimed).

Claim Rejections - 35 USC § 103

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

12. Claims 5-15 are rejected under 35 U.S. C. 103(a) as being unpatentable over Doyle in view of Sakamura et al (US005170474A), referred to hereafter as Sakamura. Doyle discloses a design which permits the services of one application to be accessed through another application as claimed above. Also like the previously mentioned claims, Doyle's design has the available services located in a list and his method makes use of the computer's memory. Doyle however fails to specify the specific role of cache and registers along with the services list.

13. In the same field of endeavor, Sakamura discloses a design that uses cache (column 4, lines 54-55, Sakamura) and registers (column 2, lines 3-8, Sakamura) (note that both cache and registers are simply types of memory as stated above) to create a queue (a queue is viewed as

being a type of a list) to store data to enhance data access times in computers (column 1, lines 60-66, Sakamura). Such a queue can be used as the list claimed to store objects or current listing of available services.

14. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura).

15. With regards to claim 5, Doyle teaches through Sakamura: that the step of creating a list of services further comprises the steps of:

- Comparing privilege data of the client program module with privilege data of objects stored in a local storage medium (Sakamura teaches how a comparison is performed between memory and the list, column 5, lines 12-55, Sakura. Memory can be a local storage medium. In addition Sakamura discloses the use of compare values, column 5, line 22, Sakura). Compare values are used as search parameters in the comparison as the privilege data is. The two are thus considered equivalent. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and

- In response to a match between privilege data of the client, recording each respective match in the list (Sakamura discloses a comparison between memory and the list where matches can occur (column 5, lines 32-41, Sakamura). When a match does occur, a notification can be sent. Since the design contains memory, it is considered obvious to one skilled in the art that the notifications can be saved, which is the same as the claimed recording of matches. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

16. With regards to claim 6, Doyle teaches through Sakamura: that the step of creating a list of services further comprises the steps of:

- Comparing a cache file with a system registry (Sakamura discloses a method by which a comparison can be made between two memories (registers are a type of memory) in search of a specified item as stated above. Sakamura further teaches how cache can be used in the design (column 4, lines 54-61, Sakamura). Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));

- Updating the cache file to reflect the system registry (Sakamura discloses a design which updates the contents of registers (column 7, lines 8-9, Sakamura). Registers like cache are simply forms of memory which can be quickly accessed; the two are thus viewed to be equivalent. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));
- Determining services in the cache file that match privilege Ids of the client program module (As previously stated, cache is a type of memory. Sakamura discloses a comparison between memory and the list where matches can occur (column 5, lines 32-41, Sakamura). When a match does occur, a notification can be sent. Since the design contains memory, it is considered obvious to one skilled in the art that the notifications can be saved, which is the same as the claimed recording of matches. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and

- Compiling a list of object locations, loader identification data, and objects in hierarchical order from the cache file (Sakamura teaches through his disclosure that a list can be created through his method (column 1, lines 60-66, Sakamura). The creation of the list involves the comparison of compare values (which can loader identification as claimed) between memory and registers (which are also a type of memory). It is well known to one in the art at the time of the invention that the location of the item on the list (the object in this case) would be known; otherwise the list would fail to offer any benefits. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

17. With regards to claim 7, Doyle teaches through Sakamura: that the step of creating a service further comprises the steps of:

- Accessing a local storage medium containing object chaining relationships and object properties (Doyle discloses a design that permits the access of objects (column 6, line 63-column 7 line 6, Doyle). Doyle's design also includes the use of "storage means" (column 8, line 28, Doyle) within the computer itself, which is a form of local memory. If the object can be accessed and the computer has local memory, then it is well known in the art at the time of the invention that the object can be located within the local memory for future access. Accordingly, it would have been obvious to one

in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and

- Constructing a container of objects forming the service based upon the object chaining relationships and object properties (Doyle's disclosure features program objects (column 6, line 63- column, line 6, Doyle). These objects are used to access the appropriate program and its services. If the program and its services can be accessed through objects, its form and function is viewed as being equivalent to the container claimed. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

18. With regards to claim 8, Doyle teaches through Sakamura: that the local storage medium comprises a cache memory device (Doyle's design features computers with local storage means as stated above. It also features RAM (column 8, lines 36-55, Doyle). RAM is memory for the purpose of providing quick load and store capabilities. It is well known to those in the art at the time of the invention that cache is a form of memory that serves to provide quick load and store capabilities. Accordingly, it would have been obvious to one in the art, at the time the invention

was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

19. With regards to claim 9, Doyle teaches through Sakamura: that the said step of constructing a container of objects further comprises the steps of:

- Determining if a master-slave relationship exists for each object in an object chain (Doyle discloses how applications can be started as child processes (column 15, line 22, Doyle). When a child process exists, a parent process must exist. This parent-child relationship is equivalent to a master-slave relationship. In addition, Doyle discloses that the method can start a child process. Doyle further states how the child process is informed with information (column 15, lines 22-38, Doyle). If information can be properly sent to a child process, means must also exist by which to detect it. Furthermore, it is well known to those skilled in the art during the time of the invention that if a child can be detected, it is not difficult for a parent to be detected. Hence, the claim is rejected. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));

- If a master-slave relationship exists for an object, then obtaining object location data and loader identification data and creating a respective slave object prior to creating a corresponding master object (Doyle discloses how child processes in his design (column 15, line 22, Doyle). If a child process exists, a parent process must exist. A parent-child process relationship is equivalent to a master-slave relationship. In addition, the creation of a new process is viewed as being equivalent to creating a new object. Furthermore, Doyle's disclosure describes how child processes can be informed about a window ID (column 15, lines 22-24, Doyle). A window ID is a type of ID and hence, the child process (slave object) should be able to handle loader ids as claimed. It is well known to those skilled in the art at the time of the invention that a slave object can serve as a master object and that a master object can serve as a slave object. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and
- If a master-slave relationship does not exist for an object, then creating the object (As stated above, a master object can be a slave object and a slave object can be a master object. In addition, an object can create another object when the need arises. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the

purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

20. With regards to claim 10, Doyle teaches through Sakamura: that steps of claim 1 further comprise of:

- Registering a service (Doyle's design describes how a registration process can be implemented to control access rights (column 12, lines 38-53, Doyle). Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and
- Updating a local memory storage medium to reflect the registering of a service (Doyle states that local memory exists in his design (column 4, line1, Doyle). The list previously stated lists the available services and it is stored within memory. Hence, when registration occurs, the list can contain updated information within memory local to the client. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of

the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

21. With regards to claim 11, Doyle teaches through Sakamura: that the step of registering a service comprises the steps of:

- Writing or deleting registry keys containing loader identification data and object location data to or from a database (The compare values of Sakamura's design that were previously stated above are can serve the role of the claimed keys and identification. They are stored within memory and hence can be written or deleted. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and
- Adding or removing files containing object data to or from a storage medium corresponding to the registry keys (Objects in Doyle's design can be stored in local memory, as mentioned above. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

22. With regards to claim 12, Doyle teaches through Sakamura: a computer –readable medium (Doyle's design describes contains computer-readable mediums such as disks and drives (column 4, line 1, Doyle)) having computer-executable instructions for performing the steps recited in claim 1 (Doyle's disclosure contains actual source code for use in implementation of the design. Source codes are computer-executable instructions. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

23. With regards to claim 13, Doyle teaches through Sakamura: a method for accessing arbitrary data content, comprising the steps of:

- Creating a service from a plurality of reusable program objects (Doyle discloses a design that creates internal object representations of the application whose services are being accessed by the application being used by the user (column 15, lines 1-8, Doyle). This creation of internal object representations is the same as creating a service. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));

- For each reusable program object, obtaining loader identification data and location data for a respective program object from a local storage medium (The loader identification data and the location data are used to locate the service to be used. Doyle's design uses program links to identify the application (whose services are) to be invoked (column 9, lines 30-32, Doyle). This can all be done through the client machine which is has storage means as stated above. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));
- Retrieving program object data from a non-local storage medium corresponding to the loader identification data and location data for the program object (Doyle describes a design that allows program objects to be retrieved from a remote computer (column 6, line 65 - column 7, line 1, Doyle). A remote computer must have storage and hence the claimed non-local storage is described. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));

- Adding the service to an existing client program module (Doyle's design allows a user to access the services of one application through another application (column 9, lines 15-45, Doyle). Such an implementation permits an enablement such as accessing a spreadsheet application through a browser application. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and
- Accessing arbitrary data content with the client program module via the service, whereby the arbitrary data content was unavailable to the existing client program module prior to adding the service (Doyle's design allows a user to access the services of a second application through a first application (column 9, lines 15-45, Doyle). Such an implementation permits an enablement such as accessing a spreadsheet application (second application) through a browser application (first application). The second application handles data that the first application would otherwise be unable to handle. Hence, as claimed, arbitrary data within the client program (first program) requires adding the service (second program) as claimed. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system

handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

24. With regards to claim 14, Doyle teaches through Sakamura: a method wherein the loader identification data and location data are passed according to one of an exchange method, a set attribute method and a get attribute method in combination with a variant parameter (Doyle's design implements the usage of Xevent inter-process communication protocol to exchange data between an application and a service being accessed (column 9, lines 6-14, Doyle). Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

25. With regards to claim 15, Doyle teaches through Sakamura: a method further comprising the steps of:

- For each object, determining if a master-slave relationship exists (Doyle discloses how applications can be started as child processes (column 15, line 22, Doyle). When a child process exists, a parent process must exist. This parent-child relationship is equivalent to a master-slave relationship. In addition, Doyle discloses that the method can start a child process. Doyle further states how the child process is informed with information (column 15, lines 22-38, Doyle). If information can be properly sent to a child process, means must also exist by which to detect it. Furthermore, it is well known to those skilled in the art during the time of the

invention that if a child can be detected, it is not difficult for a parent to be detected. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura));

- If a master-slave relationship exists for an object, then obtaining object location data and loader identification data and creating a respective slave object prior to creating a corresponding master object (Doyle discloses how child processes in his design (column 15, line 22, Doyle). If a child process exists, a parent process must exist. A parent-child process relationship is equivalent to a master-slave relationship. In addition, the creation of a new process is viewed as being equivalent to creating a new object. Furthermore, Doyle's disclosure describes how child processes can be informed about a window ID (column 15, lines 22-24, Doyle). A window ID is a type of ID and hence, the child process (slave object) should be handle loader ids as claimed. In addition, it is well known to those skilled in the art at the time of the invention that a slave object can serve as a master object and that a master object can serve as a slave object. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of

the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)); and

- If a master-slave relationship does not exist for an object, then creating the object (As stated above, a master object can be a slave object and a slave object can be a master object. In addition, an object can create another object when the need arises. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Sakamura, for the purpose of creating a method by which to access the services of applications listed in a list through another application, to improve the efficiency of the operating system handling multitask, multiuser processing, and the like (column 1, lines 64-66, Sakamura)).

26. Claims 16-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Doyle in view of Tolin et al (US005371674A), referenced as Tolin hereafter. Doyle discloses a design which permits the services of one application to be accessed through another application as claimed above. Also like the previously mentioned claims, Doyle's design has the available services located in a list and his method makes use of the computer's memory. Doyle however fails to specify the inclusion of language translation applications in his design.

27. Tolin discloses a design, which provides language translation services between many languages. The translation service performs its tasks by performing stemmer functions, lookup functions and conversion functions amongst other functions. Tolin's design is disclosed as a method and could easily be applied as an application.

28. Accordingly, it would have been obvious to one in the art, at the time the invention was made to have combined the teachings of Doyle with those of Tolin, for the purpose of creating a method by which to offer translation between two national languages (column 2, lines 59-60, Tolin).

29. With regards to claim 16, Doyle teaches through Tolin: a method for translating data from a first language to a second language, comprising the steps of:

- Requesting a list of available translation services for a client program module (Doyle discloses a design that features a "list of applications," (column 15, line 14, Doyle). The disclosure however fails to specify that translation services are offered within that list.

  Tolin however discloses a system that performs translation services (column 2, lines 59-65, Tolin). An application is a system and thus, Tolin's design can be one of the applications in Doyle's "list of applications."

  It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle));
- Creating a list of translation services available to the client program module (Doyle discloses a design that features a "user-defined list of applications," (column 15, line 14, Doyle). The fact that the list is user-defined shows that the list has to be created. The disclosure however fails to specify that translation services are offered within that list.

Tolin however discloses a system that performs translation services (column 2, lines 59-65, Tolin). An application is a system and thus, Tolin's design can be one of the applications in Doyle's "list of applications."

It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle));

- Creating a translation service from a plurality of program objects that are enumerated in the list

  (Doyle has disclosed that applications can be launched from the list. The launch of an application is equivalent to the creation of a service. The disclosure however fails to specify that translation service is one of those services that can be created.

  Tolin however discloses a system that performs translation services (column 2, lines 59-65, Tolin). An application is a system and thus, Tolin's design can be one of the applications in Doyle's "list of applications."

  It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)); and

- Translating data from the first language to a second language with a service selected from the list

(Doyle discloses a design that features a "list of applications," (column 15, line 14, Doyle). The disclosure however fails to specify that translation services are offered within that list, to enable the translating of data from a first language to a second language.

Tolin however discloses a system that performs translation services (column 2, lines 59-65, Tolin). An application is a system and thus, Tolin's design can be one of the applications in Doyle's "list of applications."

It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle));

30. With regards to claim 17, Doyle teaches through Tolin: a method wherein the step of creating a service further comprises the steps of:

- Obtaining loader identification data and location data for a program object from a local storage medium (Doyle describes a design that has the means by which to identify the application to be launched from the local user client machine (column 15, lines 18-21, Doyle). Such means are equivalent to loading id and addresses for the program/application to be used, as claimed. It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)); and

- Retrieving program object data from a non-local storage medium corresponding to the loader identification data and location data for the program object (As stated before, the client machine can access remote computers in Doyle's design, hence, non-local storage is available. The objects can be on the remote computers and can be accessed; hence a form of loader id must be available as well in the design. It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)).

31. With regards to claim 18, Doyle teaches through Tolin: a method wherein the loader identification data and location data are passed according to one of an exchange method, a set attribute method, and a get attribute method in combination with a variant parameter, whereby arbitrary data content is accessible to the client program module (Doyle's design discloses that a communication protocol is used to exchange information between the browser client and the application (column 9, lines 7-10, Doyle). This exchange is equivalent to having the claimed data passed. The communication protocol disclosed in Doyle's design is called Xevent which has attributes to set and get as claimed. It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)).

32. With regards to claim 19, Doyle teaches through Tolin: a method wherein each program object comprises one of a stemmer object, a lookup object, and a character set conversion object (Tolin discloses in his design how the translation service uses the roots of words (column 10, lines 3-6, Tolin). This deriving of the root of words is the process performed by the stemmer object and hence, the design is believed to possess the claimed stemmer object. Additionally, the task of lookups is also disclosed in the design (column 6, lines 13-14, Tolin). The lookup process in Tolin's design is the process performed by the lookup object and hence, the design is believed to contain the claimed lookup object. Finally, Tolin's disclosure describes how the design can translate between two national languages (column 2, lines 59-60, Tolin). This translation involves character set conversion and thus, the design is believed to contain the claimed character set conversion object. It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)).

33. With regards to claim 20, Doyle teaches through Tolin: a method further comprising of:

- Storing loader identification data and location data for a program object on a local storage medium and a non-local storage medium (Doyle describes a design that has the means by which to identify the application to be launched from the local user client machine (column 15, lines 18-21, Doyle). Such means are equivalent to loader ids and location data for program objects. As stated above, the design allows for a local storage medium. Also stated above, the design allows for remote computers which also contain memory, which is equivalent to a non-local storage medium. It

therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)); and

- Comparing data of a local storage medium with data on the non-local storage to determine if additional services have been added (Doyle's design describes the use of both a local client computer and a remote computer. The local client computer can access the services of the remote computer. Hence, it is well known to those skilled in the art, at the time of the invention that the local client computer can look at the contents within its memory and access the memory of the remote computer. This arrangement also permits comparison of the two storage mediums as claimed (column 6, line 63-column 7 line 6, Doyle). It therefore would have been obvious to one of ordinary skilled in the art, at the time of the invention to have the translation service taught in Tolin's disclosure with the teachings of Doyle's design to provide a method for running embedded program objects in a computer network environment (column 6, lines 50-52, Doyle)).

Conclusion

32. The prior art made of record and not relied upon is considered pertinent to the applicant's disclosure.

The following patents are cited to further show the state of the art with respect to methods and systems for accessing multiple types of electronic content in general:

US005802367A (Held et al.)

33. Any inquiries with regards to this or previous communications from the examiner should be directed towards Azizul Choudhury at the contact information listed below during the hours of 10am – 4pm, Monday-Friday.

Telephone: 703-305-7209

Email: Azizul.Choudhury@uspto.gov

34. Should he be unavailable, please contact his supervisor, David Wiley at 703-308-5221.

DAVID WILEY
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100